

## WEST Search History

[Hide Items](#)
[Restore](#)
[Clear](#)
[Cancel](#)

DATE: Saturday, January 20, 2007

Hide?	<u>Set</u> <u>Name</u>	<u>Query</u>	<u>Hit</u> <u>Count</u>
	<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=NO; OP=OR</i>		
<input type="checkbox"/>	L52	l51 and ((input\$ or entry or entries or enter or entering) near (document near table))	2
<input type="checkbox"/>	L51	l50 and (document near table)	75
<input type="checkbox"/>	L50	l49 and (character or characters or text or word or words or keyword or keywords or data or string or strings)	990
<input type="checkbox"/>	L49	l48 and table\$	990
<input type="checkbox"/>	L48	(l43 or l44 or l45 or l46 or l47) and (xml adj1 document\$)	1355
<input type="checkbox"/>	L47	715/513.ccls.	3064
<input type="checkbox"/>	L46	715/509.ccls.	212
<input type="checkbox"/>	L45	707/100.ccls.	5011
<input type="checkbox"/>	L44	707/6-7.ccls.	3030
<input type="checkbox"/>	L43	707/2.ccls.	2716
<input type="checkbox"/>	L42	l40 and (character or characters or text or word or words or keyword or keywords or data or string or strings)	2
<input type="checkbox"/>	L41	l40 and table\$	0
<input type="checkbox"/>	L40	l37 and (xml adj1 document\$)	2
<input type="checkbox"/>	L39	l38 and (document near table)	0
<input type="checkbox"/>	L38	l37 and (table near (character or characters or text or word or words or keyword or keywords or data or string or strings))	0
<input type="checkbox"/>	L37	matsuda-tomotaka.in.	8
<input type="checkbox"/>	L36	l34 and ((extract\$ or output\$ or retriev\$) near ((character or characters or text or word or words or keyword or keywords or data or string or strings) near table))	1
<input type="checkbox"/>	L35	l34 and ((input\$ or entry or entries or enter or entering) near (document near table))	1
<input type="checkbox"/>	L34	l33 and (table near (character or characters or text or word or words or keyword or keywords or data or string or strings))	41
<input type="checkbox"/>	L33	l32 and (document near table)	75
<input type="checkbox"/>	L32	((xml adj1 document) with (document or documents) with table\$ with (character or characters or text or word or words or keyword or keywords or data or string or strings))	392
<input type="checkbox"/>	L31	((xml adj1 document) with (document near table\$) with ((character or text or string\$) near table\$))	0
<input type="checkbox"/>	L30	xml.ab.	7017

10/1780,404

<input type="checkbox"/>	L29	xml.ti.	2072
<input type="checkbox"/>	L28	((document near table) with (character near string near table))	0
<input type="checkbox"/>	L27	((document near table) with (text near string near table))	0
<input type="checkbox"/>	L26	((document near table) with (text-string near table))	0
<input type="checkbox"/>	L25	((document near table) with (character-string near table))	0
	<i>DB=PGPB,USPT,USOC; PLUR=NO; OP=OR</i>		
<input type="checkbox"/>	L24	(L15 or L16) and (document near string\$)	5
<input type="checkbox"/>	L23	(L15 or L16) and (document with string\$)	24
<input type="checkbox"/>	L22	(L15 or L16) and (document with (text adj1 string\$))	4
<input type="checkbox"/>	L21	(L15 or L16) and (document near (text adj1 string\$))	0
<input type="checkbox"/>	L20	(L15 or L16) and (document near (character adj1 string\$))	4
<input type="checkbox"/>	L19	(L15 or L16) and (document with (character adj1 string\$))	14
<input type="checkbox"/>	L18	(L15 or L16) and ((document adj1 structure) with (character adj1 string\$))	2
<input type="checkbox"/>	L17	(L15 or L16) and ((document adj1 structure) with (character adj1 string\$) with (match\$ or compar\$ or similar\$ or relevanc\$ or rank\$) with (input\$ or entr\$))	0
<input type="checkbox"/>	L16	L13 and xml.ab.	54
<input type="checkbox"/>	L15	L13 and xml.ti.	24
<input type="checkbox"/>	L14	L13 and ((character adj1 string) near (entry or entries))	1
<input type="checkbox"/>	L13	L12 and (character adj1 string\$)	148
<input type="checkbox"/>	L12	L1 and (document adj1 structure)	748
<input type="checkbox"/>	L11	L8 and L10	0
<input type="checkbox"/>	L10	L1 and (document adj1 structure adj1 table)	5
<input type="checkbox"/>	L9	L1 and (document-structure adj1 table)	0
<input type="checkbox"/>	L8	L1 and (character adj1 string adj1 table)	3
<input type="checkbox"/>	L7	L1 and (character-string adj1 table)	0
<input type="checkbox"/>	L6	((character-string adj1 table) with (document-structure adj1 table))	0
<input type="checkbox"/>	L5	((character-string adj1 table) near (document-structure adj1 table))	0
<input type="checkbox"/>	L4	((character adj1 string adj1 table) near (document adj1 structure adj1 table))	0
<input type="checkbox"/>	L3	((character adj1 string adj1 table) with (document adj1 structure adj1 table))	0
<input type="checkbox"/>	L2	L1 and ((character adj1 string) near table near entr\$)	0
<input type="checkbox"/>	L1	(xml near (document or documents))	7900

END OF SEARCH HISTORY

## Dial g DataStar

[options](#)[logoff](#)[feedback](#)[help](#)[databases](#)[easy  
search](#)

## Advanced Search:

Inspec - 1898 to date (INZZ)

[limit](#)

Search history:

No.	Database	Search term	Info added since	Results	
CP		[Clipboard]		0	-
1	INZZ	xml NEAR document\$	unrestricted	1339	<a href="#">show titles</a>
2	INZZ	1 AND string NEAR (text OR character\$ OR keyword\$ OR word\$)	unrestricted	1	<a href="#">show titles</a>
3	INZZ	1 AND document NEAR table	unrestricted	2	<a href="#">show titles</a>

[hide](#) | [delete all search steps...](#) | [delete individual search steps...](#)Enter your search term(s): [Search tips](#) ☐ Thesaurus mapping whole document Information added since:  or: none   
(YYYYMMDD)[search](#)☐ Documents with images

Select special search terms from the following list(s):

- ☒ Publication year 1950-
- ☒ Publication year 1898-1949
- ☒ Inspec thesaurus - browse headings A-G
- ☒ Inspec thesaurus - browse headings H-Q
- ☒ Inspec thesaurus - browse headings R-Z
- ☒ Inspec thesaurus - enter a term
- ☒ Classification codes A: Physics, 0-1
- ☒ Classification codes A: Physics, 2-3
- ☒ Classification codes A: Physics, 4-5
- ☒ Classification codes A: Physics, 6
- ☐ Classification codes A: Physics, 7

10/780,404

# Dial g DataStar

options

logoff

feedback

help

databases

search  
page

titles

## Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

save

locally as: PDF document



search strategy: do not include the search strategy



order

copy to  
Clipboard
☒ Select All

1 Converting relational database into XML document.

2 99 tricks to amaze your friends and impress your boss with the DOM and XML in W

☒ document 1 of 2 Order Document

Inspec - 1898 to date (INZZ)

### Accession number & update

0007092967 20070101.

### Title

Converting relational database into **XML document.**

### Conference information

12th International Workshop on Database and Expert Systems Applications, Munich, Germany, 3-7 Sept. 2001.

### Source

12th International Workshop on Database and Expert Systems Applications, 2001, p. 61-5, 9 refs, pp. xxv+970, ISBN: 0-7695-1230-5. Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA.

### Author(s)

Fong-J, Pang-F, Bloor-C.

Editor(s): Tjoa-A-M, Wagner-R-R.

### Author affiliation

Fong, J., Pang, F., Dept. of Comput. Sci., City Univ. of Hong Kong, China.

### Abstract

**XML** (eXtensible Markup Language) has emerged and is gradually being accepted as the standard for data interchange in the Internet world. **XML** databases are packaged by the key relational database vendors in the market as the extender or cartridge to the relational database management system. Interoperation of relational database and **XML** database involves schema and data translations. The paper provides a methodology of translating the conceptual schema of a relational database into **XML** schema through EER (extended entity relationship) model. Physical data are then translated from relational **table** to **XML document**. The semantics of the relational database, captured in EER diagram, are mapped to **XML** schema using stepwise procedures. The physical data are then mapped to an **XML document** under the definitions of the **XML** schema.

### Descriptors

ELECTRONIC-DATA-INTERCHANGE; ENTITY-RELATIONSHIP-MODELLING;  
HYPERMEDIA-MARKUP-LANGUAGES; INTERNET; RELATIONAL-DATABASES.

### Classification codes

C6160D Relational-databases\*;

C6110 Systems-analysis-and-programming;

C6130D Document-processing-techniques;  
C6130E Data-interchange;  
C6130M Multimedia;  
C6140D High-level-languages;  
C7210N Information-networks.

**Keywords**

relational-database-conversion; **XML-document**; eXtensible-Markup-Language; **XML-standard**; data-interchange; Internet; **XML-databases**; relational-database-vendors; relational-database-management-system; schema-translations; data-translations; conceptual-schema; **XML-schema**; extended-entity-relationship-model; **relational-table**; relational-database; EER-diagram; stepwise-procedures; physical-data.

**Treatment codes**

P Practical.

**Language**

English.

**Publication type**

Conference-paper.

**Availability**

CCCC: 0-7695-1230-5/01/\$10.00.

**Digital object identifier**

10.1109/DEXA.2001.953042.

**Publication year**

2001.

**Publication date**

20010000.

**Edition**

2001045.

**Copyright statement**

Copyright 2001 IEE.

(c) 2007 The Institution of Engineering and Technology

---

☒ **document 2 of 2** Order Document

**Inspec - 1898 to date (INZZ)**

**Accession number & update**

0006488963 20070101.

**Title**

99 tricks to amaze your friends and impress your boss with the DOM and **XML** in Web browsers.

**Conference information**

Proceedings of **XML** EUROPE '99, Granada, Spain, 26-30 April 1999.

Sponsor(s): OASIS; W3C World Wide Web Consortium.

**Source**

**XML** Europe '99 Conference Proceedings, 1999, p. 335-56, 0 refs, pp. ix +759.

Publisher: Graphic Commun. Assoc, Alexandria, VA, USA.

**Author(s)**

Leventhal-M.

**Author affiliation**

Leventhal, M., CITEC, Vaasa, Finland.

**Abstract**

The client-side capability of the Web is about to change and improve dramatically with the introduction of the DOM (**Document** Object Model) and **XML** in Web browsers. This paper shows what can be done and shows you how to do it. The paper contains walk throughs of code as well as a high level description of the processing which should be of interest to **XML** Web developers. Three DOM applications are presented in this paper: a rapid **document** structure viewer, a **document** annotation tool, and a **table** which can be sorted on any column. The last example compares DOM and XSL (Extensible Stylesheet Language) approaches to this problem, arguing for the advantages of the DOM

in terms of superior capability and comprehensibility of the code.

**Descriptors**

 DISTRIBUTED-OBJECT-MANAGEMENT;  HYPERMEDIA-MARKUP-LANGUAGES;  
 INFORMATION-RESOURCES;  ONLINE-FRONT-ENDS.

**Classification codes**

C7250N Search-engines\*;  
C6150N Distributed-systems-software;  
C6130M Multimedia;  
C6130D Document-processing-techniques;  
C6140D High-level-languages.

**Keywords**

DOM; **XML**; Web-browsers; **Document-Object-Model**; World-Wide-Web-Consortium; high-level-descriptions; **document-structure-viewer**; **document-annotation-tool**; Extensible-Stylesheet-Language; code- comprehensibility.

**Treatment codes**

P Practical.

**Language**

English.

**Publication type**

Conference-paper.

**Publication year**

1999.

**Publication date**

19990000.

**Edition**

2000005.

**Copyright statement**

Copyright 2000 IEE.

(c) 2007 The Institution of Engineering and Technology

save

locally as: PDF document



search strategy:

do not include the search strategy



order

copy to  
Clipboard

Top - News & FAQs - Dialog

© 2007 Dialog


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

retrieval and character string and table and xml document

SEARCH


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used

retrieval and character string and table and xml document

Found 48,976 of 196,780

Sort results by

relevance


[Save results to a Binder](#)
[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Display results

expanded form


[Search Tips](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐ ☐ ☐ ☐ ☐

### 1 [A compressor for effective archiving, retrieval, and updating of XML documents](#)



Jun-Ki Min, Myung-Jae Park, Chin-Wan Chung

 August 2006 **ACM Transactions on Internet Technology (TOIT)**, Volume 6 Issue 3

Publisher: ACM Press

Full text available: pdf(1.30 MB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Like HTML, many XML documents are resident on native file systems. Since XML data is irregular and verbose, the disk space and the network bandwidth are wasted. To overcome the verbosity problem, research on compressors for XML data has been conducted. Some XML compressors do not support querying compressed data, while other XML compressors which support querying compressed data blindly encode tags and data values using predefined encoding methods. Existing XML compressors do not provide the fac ...

**Keywords:** Compression, XML, query processing

### 2 [XRel: a path-based approach to storage and retrieval of XML documents using relational databases](#)


 August 2001 **ACM Transactions on Internet Technology (TOIT)**, Volume 1 Issue 1

Publisher: ACM Press

Full text available: pdf(264.27 KB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This article describes XRel, a novel approach for storage and retrieval of XML documents using relational databases. In this approach, an XML document is decomposed into nodes on the basis of its tree structure and stored in relational tables according to the node type, with path information from the root to each node. XRel enables us to store XML documents using a fixed relational schema without any information about DTDs and also to utilize indices such as the B+

**Keywords:** XML query, XPath, text markup, text tagging

### 3 [Integrating document and data retrieval based on XML](#)




Jan-Marco Bremer, Michael Gertz

 January 2006 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 15 Issue 1

10/780, 4024

**Publisher:** Springer-Verlag New York, Inc.

Full text available:  pdf(841.10 KB) Additional Information: [full citation](#), [abstract](#)

For querying structured and semistructured data, data retrieval and document retrieval are two valuable and complementary techniques that have not yet been fully integrated. In this paper, we introduce integrated information retrieval (IIR), an XML-based retrieval approach that closes this gap. We introduce the syntax and semantics of an extension of the XQuery language called XQuery/IR. The extended language realizes IIR and thereby allows users to formulate new kinds of queries by nesting rank ...

**Keywords:** Data retrieval, Document retrieval, Index structures, Integrated information retrievals, Structural join, XML

#### 4 An automated approach for retrieving hierarchical data from HTML tables



Seung-Jin Lim, Yiu-Kai Ng

November 1999 **Proceedings of the eighth international conference on Information and knowledge management CIKM '99**

**Publisher:** ACM Press

Full text available:  pdf(1.74 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Among the HTML elements, HTML tables [RHJ98] encapsulate hierarchically structured data (hierarchical data in short) in a tabular structure. HTML tables do not come with a rigid schema and almost any forms of two-dimensional tables are acceptable according to the HTML grammar. This relaxation complicates the process of retrieving hierarchical data from HTML tables. In this paper, we propose an automated approach for retrieving hierarchical data from HTML tables. The proposed approach constr ...

#### 5 Efficiently publishing relational data as XML documents

Jayavel Shanmugasundaram, Eugene Shekita, Rimon Barr, Michael Carey, Bruce Lindsay, Hamid Pirahesh, Berthold Reinwald

September 2001 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 10 Issue 2-3

**Publisher:** Springer-Verlag New York, Inc.

Full text available:  pdf(216.67 KB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

XML is rapidly emerging as a standard for exchanging business data on the World Wide Web. For the foreseeable future, however, most business data will continue to be stored in relational database systems. Consequently, if XML is to fulfill its potential, some mechanism is needed to publish relational data as XML documents. Towards that goal, one of the major challenges is finding a way to efficiently structure and tag data from one or more tables as a hierarchical XML document. Different alterna ...

**Keywords:** Publishing, Relational databases, XML

#### 6 Document creation I: Creating structured PDF files using XML templates



Matthew R. B. Hardy, David F. Brailsford, Peter L. Thomas

October 2004 **Proceedings of the 2004 ACM symposium on Document engineering DocEng '04**

**Publisher:** ACM Press

Full text available:  pdf(166.87 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes a tool for recombining the logical structure from an XML document with the typeset appearance of the corresponding PDF document. The tool uses the XML representation as a template for the insertion of the logical structure into the existing PDF



document thereby creating a Structured/Tagged PDF. The addition of logical structure adds value to the PDF in three ways: the accessibility is improved (PDF screen readers for visually impaired users perform better) media options a ...

**Keywords:** PDF, XML, logical structure insertion

## 7 Efficient filtering of XML documents with XPath expressions

C.-Y. Chan, P. Felber, M. Garofalakis, R. Rastogi

December 2002 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 11 Issue 4

**Publisher:** Springer-Verlag New York, Inc.

Full text available:  [pdf\(383.34 KB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

The publish/subscribe paradigm is a popular model for allowing publishers (i.e., data generators) to selectively disseminate data to a large number of widely dispersed subscribers (i.e., data consumers) who have registered their interest in specific information items. Early publish/subscribe systems have typically relied on simple subscription mechanisms, such as keyword or "bag of words" matching, or simple comparison predicates on attribute values. The emergence of XML as a standard ...

**Keywords:** Data dissemination, Document filtering, Index structure, XML, XPath


## 8 XIRQL: An XML query language based on information retrieval concepts



Norbert Fuhr, Kai Großjohann

April 2004 **ACM Transactions on Information Systems (TOIS)**, Volume 22 Issue 2

**Publisher:** ACM Press

Full text available:  [pdf\(281.91 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

XIRQL ("circle") is an XML query language that incorporates imprecision and vagueness for both structural and content-oriented query conditions. The corresponding uncertainty is handled by a consistent probabilistic model. The core features of XIRQL are (1) document ranking based on index term weighting, (2) specificity-oriented search for retrieving the most relevant parts of documents, (3) datatypes with vague predicates for dealing with specific types of content and (4) structural vagueness f ...

**Keywords:** Path algebra, XML, XQuery, probabilistic retrieval, ranked retrieval, vague predicates

## 9 An analysis of XML database solutions for the management of MPEG-7 media descriptions



Utz Westermann, Wolfgang Klas

December 2003 **ACM Computing Surveys (CSUR)**, Volume 35 Issue 4

**Publisher:** ACM Press

Full text available:  [pdf\(448.76 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

MPEG-7 constitutes a promising standard for the description of multimedia content. It can be expected that a lot of applications based on MPEG-7 media descriptions will be set up in the near future. Therefore, means for the adequate management of large amounts of MPEG-7-compliant media descriptions are certainly desirable. Essentially, MPEG-7 media descriptions are XML documents following media description schemes defined with a variant of XML Schema. Thus, it is reasonable to investigate curren ...

**Keywords:** MPEG-7, XML database systems, multimedia databases

10 Industrial session: XML support in relational system: Native XML support in DB2 universal database

Matthias Nicola, Bert van der Linden


August 2005 **Proceedings of the 31st international conference on Very large data bases VLDB '05**

**Publisher:** VLDB Endowment

Full text available:  pdf(240.25 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The major relational database systems have been providing XML support for several years, predominantly by mapping XML to existing concepts such as LOBs or (object-) relational tables. The limitations of these approaches are well known in research and industry. Thus, a forthcoming version of DB2 Universal Database® is enhanced with comprehensive *native* XML support. "Native" means that XML documents are stored on disk pages in tree structures matching the XML data model. This avoids the ...

11 XML: Compressing and searching XML data via two zips

 P. Ferragina, F. Luccio, G. Manzini, S. Muthukrishnan

May 2006 **Proceedings of the 15th international conference on World Wide Web WWW '06**


**Publisher:** ACM Press

Full text available:  pdf(314.30 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

XML is fast becoming the standard format to store, exchange and publish over the web, and is getting embedded in applications. Two challenges in handling XML are its size (the XML representation of a document is significantly larger than its native state) and the complexity of its search (XML search involves path and content searches on labeled tree structures). We address the basic problems of compression, navigation and searching of XML documents. In particular, we adopt recently proposed theo ...

**Keywords:** XML compression and indexing, labeled trees

12 Document searching, document annotation, and document metadata: Prefiltering techniques for efficient XML document processing

 Chia-Hsin Huang, Tyng-Ruey Chuang, Hahn-Ming Lee

November 2005 **Proceedings of the 2005 ACM symposium on Document engineering DocEng '05**

**Publisher:** ACM Press

Full text available:  pdf(442.96 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Document Object Model (DOM) and Simple API for XML (SAX) are the two major programming models for XML document processing. Each, however, has its own efficiency limitation. DOM assumes an in-core representation of XML documents which can be problematic for large documents. SAX needs to scan over the document in a linear manner in order to locate the interesting fragments. Previously, we have used tree-to-table mapping and indexing techniques to help answer structural queries to large, or large C....

**Keywords:** DOM, SAX, prefiltering, structural query, two-phased XML processing model

13 Industrial, applications, and experience sessions: Industry 5: Tools and experience:

On the path to efficient XML queries

Andrey Balmin, Kevin S. Beyer, Fatma Özcan, Matthias Nicola

September 2006 **Proceedings of the 32nd international conference on Very large data bases - Volume 32 VLDB'2006****Publisher:** VLDB EndowmentFull text available:  pdf(562.96 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

XQuery and SQL/XML are powerful new languages for querying XML data. However, they contain a number of stumbling blocks that users need to be aware of to get the expected results and performance. For example, certain language features make it hard if not impossible to exploit XML indexes. The major database vendors provide XQuery and SQL/XML support in their current or upcoming product releases. In this paper, we identify common pitfalls gleaned from the experiences of early adopters of this func ...

14 Paper session IR-1 (information retrieval): XML retrieval: Structured queries in XML retrieval

Jaap Kamps, Maarten Marx, Maarten de Rijke, Börkur Sigurbjörnsson

October 2005 **Proceedings of the 14th ACM international conference on Information and knowledge management CIKM '05****Publisher:** ACM PressFull text available:  pdf(260.03 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Document-centric XML is a mixture of text and structure. With the increased availability of document-centric XML content comes a need for query facilities in which both structural constraints and constraints on the content of the documents can be expressed. How does the expressiveness of languages for querying XML documents help users to express their information needs? We address this question from both an experimental and a theoretical point of view. Our experimental analysis compares a struct ...

**Keywords:** XML retrieval, XPath, full-text XML querying15 Preparing heterogeneous XML for full-text search

Miro Lehtonen

October 2006 **ACM Transactions on Information Systems (TOIS)**, Volume 24 Issue 4**Publisher:** ACM PressFull text available:  pdf(228.25 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


XML retrieval is facing new challenges when applied to heterogeneous XML documents, where next to nothing about the document structure can be taken for granted. We have developed solutions where some of the heterogeneity issues are addressed. Our fragment selection algorithm selectively divides a heterogeneous document collection into equi-sized fragments with full-text content. If the content is considered too data-oriented, it is not accepted. The algorithm needs no information about element n ...

**Keywords:** XML retrieval, heterogeneous documents, indexing16 XML: XML screamer: an integrated approach to high performance XML parsing, validation and deserialization

Margaret G. Kostoulas, Morris Matsa, Noah Mendelsohn, Eric Perkins, Abraham Heifets, Martha Mercaldi

May 2006 **Proceedings of the 15th international conference on World Wide Web WWW '06****Publisher:** ACM Press

Full text available: Additional Information:

 pdf(303.88 KB)

[full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes an experimental system in which customized high performance XML parsers are prepared using parser generation and compilation techniques. Parsing is integrated with Schema-based validation and deserialization, and the resulting validating processors are shown to be as fast as or in many cases significantly faster than traditional nonvalidating parsers. High performance is achieved by integration across layers of software that are traditionally separate, by avoiding unnecessary ...


**Keywords:** JAX-RPC, SAX, XML, XML schema, parsing, performance, schema compilation, validation

### 17 [Data transformation and duplicate detection: Detecting duplicate objects in XML documents](#)

 Melanie Weis, Felix Naumann

June 2004 **Proceedings of the 2004 international workshop on Information quality in information systems IQIS '04**

**Publisher:** ACM Press


Full text available:  pdf(1.03 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

The problem of detecting duplicate entities that describe the same real-world object (and purging them) is an important data cleansing task, necessary to improve data quality. For data stored in a flat relation, numerous solutions to this problem exist. As XML becomes increasingly popular for data representation, algorithms to detect duplicates in nested XML documents are required. In this paper, we present a domain-independent algorithm that effectively identifies duplicates in an XML document. ...

**Keywords:** XML, data cleansing, duplicate detection, similarity

### 18 [Path sharing and predicate evaluation for high-performance XML filtering](#)

 Yanlei Diao, Mehmet Altinel, Michael J. Franklin, Hao Zhang, Peter Fischer

December 2003 **ACM Transactions on Database Systems (TODS)**, Volume 28 Issue 4

**Publisher:** ACM Press


Full text available:  pdf(543.40 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

XML filtering systems aim to provide fast, on-the-fly matching of XML-encoded data to large numbers of query specifications containing constraints on both structure and content. It is now well accepted that approaches using event-based parsing and Finite State Machines (FSMs) can provide the basis for highly scalable structure-oriented XML filtering systems. The XFilter system [Altinel and Franklin 2000] was the first published FSM-based XML filtering approach. XFilter used a separate FSM per pa ...


**Keywords:** Nondeterministic Finite Automaton, XML filtering, content-based matching, nested path expressions., path sharing, predicate evaluation, structure matching

### 19 [Document analysis and reconstruction: Recognizing records from the extracted cells of microfilm tables](#)

 Kenneth M. Tubbs, David W. Embley

November 2002 **Proceedings of the 2002 ACM symposium on Document engineering DocEng '02**

**Publisher:** ACM Press

Full text available:  pdf(1.50 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Microfilm documents contain a wealth of information, but extracting and organizing this information by hand is slow, error-prone, and tedious. As an initial step toward automating access to this information, we describe in this paper an algorithmic process to automatically identify record patterns found in microfilm tables for pre-specified application domains. Our table-processing algorithm accepts an XML input file describing the individual cells of a table taken from a microfilm document, and ...

**Keywords:** automated recognition of record patterns, geometric layout, microfilm tables, ontology matching

## 20 Burst tries: a fast, efficient data structure for string keys



Steffen Heinz, Justin Zobel, Hugh E. Williams

April 2002 **ACM Transactions on Information Systems (TOIS)**, Volume 20 Issue 2

**Publisher:** ACM Press

Full text available: [pdf\(324.84 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Many applications depend on efficient management of large sets of distinct strings in memory. For example, during index construction for text databases a record is held for each distinct word in the text, containing the word itself and information such as counters. We propose a new data structure, the burst trie, that has significant advantages over existing options for such applications: it uses about the same memory as a binary search tree; it is as fast as a trie; and, while not as fast as a ...

**Keywords:** Binary trees, splay trees, string data structures, text databases, tries, vocabulary accumulation

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)